

Software



APAX-5522PE

Embedded Linux

User Manual

ADVANTECH

Enabling an Intelligent Planet

Copyright

The documentation and the software included with this product are copyrighted 2012 by Advantech Co., Ltd. All rights are reserved. Advantech Co., Ltd. reserves the right to make improvements in the products described in this manual at any time without notice. No part of this manual may be reproduced, copied, translated or transmitted in any form or by any means without the prior written permission of Advantech Co., Ltd. Information provided in this manual is intended to be accurate and reliable. However, Advantech Co., Ltd. assumes no responsibility for its use, nor for any infringements of the rights of third parties, which may result from its use.

About This Manual

Thank you for using Advantech products. This manual is a supplementation to APAX-5522 user manual, and it emphasizes on Linux for APAX-5522, other detail information about APAX-5522 hardware, jumper setting, serial port connection and so on, please refer to the APAX-5522 Hardware manual.

Edition 1
October 2012

Contents

Chapter 1 APAX-5522 features overview 1

1.1	Hardware features.....	2
1.2	Software features	2

Chapter 2 APAX-5522 embedded Linux configuration and usage 3

2.1	Basic Linux command	4
2.1.1	ls: list the file information.....	4
2.1.2	cp: copy file or directory	4
2.1.3	mv: move or rename file/directory.....	4
2.1.4	rm: delete file or directory	4
2.1.5	mkdir: create directory	4
2.1.6	rmdir: remove directory	4
2.1.7	pwd: show the current path.....	4
2.1.8	chmod: change file mod bit.....	5
2.1.9	./ : execution of your program	5
2.1.10	ps : report a snapshot of the current process	5
2.1.11	kill -9 PID: kill the process.....	5
2.1.12	cd: change directory.....	5
2.1.13	clear: clear the screen.....	5
2.1.14	reboot: reboot the APAX-5522.....	5
2.2	Connecting the APAX-5522 Linux.....	6
2.3	Configure IP	7
2.4	Configure RTC	8
2.5	Serial port.....	9
2.6	Mount the CF card	9
2.7	Mount the USB storage device	9
2.8	Telnet service.....	9
2.9	FTP service	10

Chapter 3 Development guide 13

3.1	Establish develop environment	14
3.2	Quick start to develop application	14
3.2.1	Compile HelloWorld.c	15
3.2.2	Upload “HelloWorld” to the APAX-5522	15
3.2.3	Execute “HelloWorld” on the APAX-5522	16
3.2.4	Sample Makefile of HelloWorld.c.....	16

Chapter 4 APAX utility tool kit for Linux 19

4.1	Overview	20
4.2	Command introduction	21
4.2.1	Display APAX utility information.....	21
4.2.2	Get module list	21
4.2.3	Get module information.....	22
4.2.4	Get channel information.....	24
4.2.5	Set channel value	28
4.2.6	Set channel range.....	33
4.2.7	Set DO module configuration.....	35

4.2.8	Set AO module configuration.....	37
4.2.9	Set CNT module configuration.....	39
4.2.10	Get the APAX-5017PE AI module with timestamp	44
4.2.11	Get the APAX-5040PE DI module with timestamp	47
A.1	ADAM/APAX .NET Utility General Window	50

Appendix A

APAX 5000 I/O module range code settings 49

Chapter 1

**APAX-5522 features
overview**

1.1 Hardware features

- CPU: XScale PXA270, 520 MHz
- Memory: Flash 32M bytes, SDRAM 64M bytes
- Diagnostics LEDs: 1xPower, 1xRun, 1xError, 1xBattery, 2xLAN link/activity/speed
- Real-time Clock: Yes
- Watchdog Timer: Yes
- Power Consumption: 5 W @ 24 VDC (typical)
- USB Ports: 1 (USB 1.1)
- LAN Ports: 2 (10/100 Mbps)
- Serial Ports: 2 (RS-232)
- Serial Baud Rate: 9600 ~ 115200 bps
- Storage: 1 x Type II Compact Flash card slot (internal)

1.2 Software features

- Boot loader: U-boot-1.1.6
- Kernel version: Linux 2.6.25.8-rt7
- C library: glibc version: 2.8
- OS boot up time: <= 40 seconds
- Flash partition:

Boot loader	512k
Kernel	2.3 M
Root file system	25.8 M
- Root files system: JFFS2

Chapter 2

**APAX-5522 embedded
Linux configuration
and usage**

2.1 Basic Linux command

2.1.1 ls: list the file information

Example:
ls -l
list detailed information of file.
ls -a
list all files including hidden files.
ls -t
list the files that are arranged by time.

2.1.2 cp: copy file or directory

Example:
cp test.c /home/user/
copy test.c file to /home/user/
cp -a mywork /home/user/
copy mywork directory to /home/user/

2.1.3 mv: move or rename file/directory

Example:
mv test.c main.c
rename file test.c to main.c
#mv workspace/ backup/
rename directory workspace as backup

2.1.4 rm: delete file or directory

Example:
rm test.c
delete file test.c
rm -r workspace
delete directory workspace

2.1.5 mkdir: create directory

Example:
mkdir workspace
create workspace directory

2.1.6 rmdir: remove directory

Example:
rmdir workspace
remove workspace directory

2.1.7 pwd: show the current path

Example:
pwd
/home/user

2.1.8 chmod: change file mod bit

Example:

```
# chmod 754 test
# ls -l test
-rwxr-xr-- 1 adv adv 8004 Jan 1 1970 test
```

A combination of the letters controls which user can access to the file.

754 means: 111(read, write, execute) 101(read, write, execute) 100(read, write, execute).
The first number 7 is owner can read and write and execute file. The second number 5 is group can only read and execute file. The third number 4 is all users can only read file.

2.1.9 ./ : execution of your program

Example:

```
# ls -l test
-rwxr-xr-- 1 adv adv 8004 Jan 1 1970 test

# ./test
This is my test program!!
```

2.1.10 ps : report a snapshot of the current process

Example:

```
# ps
 PID  Uid   VmSize Stat Command
  1 root    544 S  init [5]
  2 root     SW< [kthreadd]
  3 root     SW< [ksoftirqd/0]
  4 root     SW< [watchdog/0]
  5 root     SW< [events/0]
  6 root     SW< [khelper]
 ...
 ...
1309 root    904 S  -sh
1571 root    896 S  sh
1593 root    320 R  ./test
1594 root    728 R  ps
```

2.1.11 kill -9 PID: kill the process

Example:

```
# kill -9 1593
kill the running test program. the PID is 1593
```

2.1.12 cd: change directory

2.1.13 clear: clear the screen

2.1.14 reboot: reboot the APAX-5522

2.2 Connecting the APAX-5522 Linux

Users can check the RUN LED lights when the APAX-5522 is powered on. Then can use Telnet to connect to the APAX-5522's console if you have known the IP address. Once you connect to the APAX-5522 you can proceed to configure network settings, as described in the next section. Telnet can be used locally by using a crossover Ethernet cable to connect your computer to the APAX-5522, or over a LAN or the Internet. The default IP address and netmask are shown below. To login, type the Login name and password as requested. The default login name is root and no password.

	Default IP Address	Default netmask
LAN1(eth0)	10.0.0.1	255.255.255.0
LAN2(eth1)	10.0.0.2	255.255.255.0

```
# ifconfig
eth0      Link encap:Ethernet HWaddr 00:D0:C9:F0:49:91
           inet addr:10.0.0.1 Bcast:10.0.0.255 Mask:255.255.255.0
           UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
           RX packets:1139 errors:0 dropped:0 overruns:0 frame:0
           TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1000
           RX bytes:132363 (129.2 KiB) TX bytes:0 (0.0 B)
           Interrupt:157 Base address:0xc000

eth1      Link encap:Ethernet HWaddr 00:D0:C9:F0:49:92
           inet addr:10.0.0.2 Bcast:10.0.0.255 Mask:255.255.255.0
           UP BROADCAST MULTICAST MTU:1500 Metric:1
           RX packets:0 errors:0 dropped:0 overruns:0 frame:0
           TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1000
           RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
           Interrupt:158

lo        Link encap:Local Loopback
           inet addr:127.0.0.1 Mask:255.0.0.0
           UP LOOPBACK RUNNING MTU:16436 Metric:1
           RX packets:0 errors:0 dropped:0 overruns:0 frame:0
           TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:0
           RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

# _
```

In addition, we also provide a PC-side client tool that users can use it to get the IP address of all the APAX-5522 devices. Please plug the network cable to the LAN (eth0) port of the APAX-5522 devices.



2.3 Configure IP

Users can configure IP and change LAN physical address in “/etc/rc”, OS then will execute the shell script automatically when booting up. The default setting for Ethernet is static IP address.

```
ifconfig lo 127.0.0.1
ifconfig eth0 10.0.0.1 netmask 255.255.255.0
ifconfig eth1 10.0.0.2 netmask 255.255.255.0
ip rule add from 10.0.0.1 lookup 10
ip route add table 10 default src 10.0.0.1 dev eth0
ip rule add from 10.0.0.2 lookup 11
ip route add table 11 default src 10.0.0.2 dev eth1
route add default gw 10.0.0.254 eth1
route add default gw 10.0.0.254 eth0
inetd &
cat /etc/motd
```

If user wants to get the eth0 IP by dhcp, please mask and modify the “/etc/rc” file as follows:

```
ifconfig lo 127.0.0.1
if [ -e /var/run/dhcpcd-eth0.pid ]; then
rm -f /var/run/dhcpcd-eth0.pid
fi
if [ -e /var/run/dhcpcd-eth1.pid ]; then
rm -f /var/run/dhcpcd-eth1.pid
fi
dhcpcd eth0 &
dhcpcd eth1 &
inetd &
cat /etc/motd
```

Sometimes, you may need to modify the IP routing table by manually that can route to the correct LAN port out. Please refer to the following case, and manually modify the rules of the IP routing table when the system startup. The red color words depend on the routing table. You can type “route” instruction to know the current setting.

```
# route
Kernel IP routing table
Destination     Gateway     Genmask      Flags Metric Ref Use Iface
10.0.0.0        *          255.255.255.0 U     0      0      0 eth0
10.0.0.0        *          255.255.255.0 U     0      0      0 eth1
default         10.0.0.254  0.0.0.0      UG    0      0      0 eth0
default         10.0.0.254  0.0.0.0      UG    0      0      0 eth1
#
```

For example, I just use LAN2 case therefore, I need to remove the routing rules of the LAN1(eth0) from the routing table. Type following instructions:

```
# route del -net 10.0.0.0 netmask 255.255.255.0 dev eth0
# route del default gw 10.0.0.254 eth0
```

Now I can route to the correct LAN port out.

```
# route
Kernel IP routing table
Destination     Gateway      Genmask      Flags Metric Ref  Use Iface
10.0.0.0        *           255.255.255.0 U     0      0      0 eth1
default         10.0.0.254  0.0.0.0      UG    0      0      0 eth1
#
```

■ IP routing table rules

System startup	Typing the Linux instructions
Just use LAN1 (eth0)	route del -net 10.0.0.0 netmask 255.255.255.0 dev eth1 route del default gw 10.0.0.254 eth1 [Note] If you reinsert the network cable to LAN2, please type the following commands to create the LAN1 routing table rules. route add -net 10.0.0.0 netmask 255.255.255.0 dev eth1 route add default gw 10.0.0.254 eth1
Just use LAN2 (eth1)	route del -net 10.0.0.0 netmask 255.255.255.0 dev eth0 route del default gw 10.0.0.254 eth0 [Note] If you reinsert the network cable to LAN1, please type the following commands to create the LAN1 routing table rules. route add -net 10.0.0.0 netmask 255.255.255.0 dev eth0 route add default gw 10.0.0.254 eth0
Both use LAN1&LAN2	No need to modify

2.4 Configure RTC

You may use the “date” command to print or set the system date.

SYNOPSIS:

```
date MMDDhhmmYYYY.SS
```

MM = Month

DD = Date

hhmm = hour and minute

YYYY = Year

SS = second

For example, Fri Apr 03 18:54:12 UTC 2009

```
# date 040318542009.12
Fri Apr 3 15:54:12 UTC 2009
```

In order to maintain the correct time when powering off, users should use the hwclock to write the current time to real time clock.

```
# hwclock --systohc
```

Now if you reboot the APAX-5522, the time is still correct.

2.5 Serial port

APAX-5522 has two serial ports; the device node is /dev/ttyAP0 and /dev/ttyAP1. For serial port test, APAX-5522 has built-in test program called st (serialtest) for user test serial port functionality. Please see below example:

Use serial port to receive data at the Baud Rate of 115200.

```
#st /dev/ttyAP0 -r -b 115200
```

Use serial port to send data at the Baud Rate of 115200.

```
#st /dev/ttyAP0 -s -b 115200
```

2.6 Mount the CF card

CF card is usually identified as hard disk, the device is /dev/mmcblk0, and the user can use the mount command to mount the CF card to a folder. For example:

```
#mount /dev/hda1 /mnt
```

The system now support ext2, ext3 and fat32 file system. If your CF card's file system is FAT32, you should mount it as follow

```
#mount -t vfat /dev/hda1 /mnt
```

2.7 Mount the USB storage device

Users also can mount the USB storage device to the APAX-5522, before they can access the USB storage device. The steps are as follows:

List the partition tables for the specified devices and then exit.

```
#fdisk -l
```

USB storage device is usually identified as SCSI device, the device is /dev/sda1, and the user can use the mount command to mount the USB device to a folder.

Mount USB Storage device

```
#mount /dev/sda1 /mnt
```

Unmount USB Storage device

```
#umount /mnt
```

2.8 Telnet service

When you boot up the APAX-5522, the telnet service is already started by default. The users can telnet to APAX-5522 by telnet client in another computer. When you first telnet the APAX-5522, you should create a user and set password, then you can use the user as username and the password you set as password.

- Create user

You may use the adduser command to create user account and password.

```
172.19.1.108
# adduser adv
Changing password for adv
Enter the new password (minimum of 5, maximum of 8 characters)
Please use a combination of upper and lower case letters and numbers.
Enter new password:
Re-enter new password:
Password changed.
#
```

After setting, you can telnet to APAX-5522 by new user account.

```
172.19.1.108
Linux 2.6.25.8-rt7 (172.19.1.67) (ttyp0)

login: adv
Password:
#
```

- Remove user

Use root id login to APAX-5522, The steps are as follows:

1. Delete user
 # deluser adv
2. Remove user home directory
 # rm -r /home/adv

2.9 FTP service

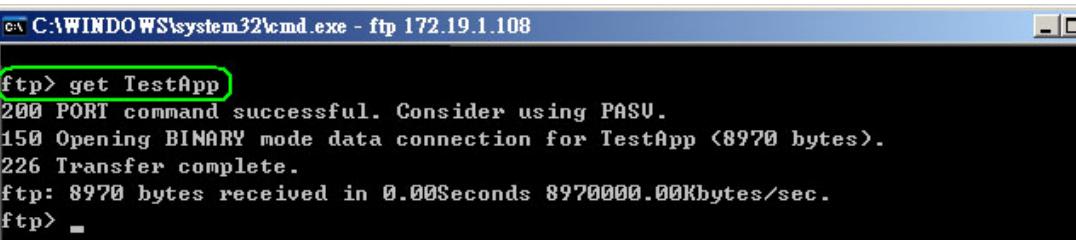
When you boot up the APAX-5522, the ftp service is already started by default. The user can use the ftp client to connect APAX-5522, then download files from APAX-5522 or upload files to APAX-5522 through network.

Example:

- Upload file to APAX-5522

```
C:\WINDOWS\system32\cmd.exe - ftp 172.19.1.108
C:\>ftp>ftp 172.19.1.108
Connected to 172.19.1.108.
220 <vsFTPd 2.3.4>
User <172.19.1.108:<none>>: adv
331 Please specify the password.
Password:
230 Login successful.
ftp> bin
200 Switching to Binary mode.
ftp> put HelloWorld
200 PORT command successful. Consider using PASV.
150 Ok to send data.
226 Transfer complete.
ftp: 8970 bytes sent in 0.00Seconds 8970000.00Kbytes/sec.
ftp>
```

- Download file from APAX-5522



```
C:\WINDOWS\system32\cmd.exe - ftp 172.19.1.108
ftp> get TestApp
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for TestApp (8970 bytes).
226 Transfer complete.
ftp: 8970 bytes received in 0.00Seconds 8970000.00Kbytes/sec.
ftp> -
```

Basic FTP command:

!: exit FTP back to pc temporarily
bin: transfer files in "Binary" mode
ascii: transfer files in "ASCII" mode
get: download file from APAX-5522
put: upload file from PC to APAX-5522
bye: exit FTP

Chapter 3

Development guide

3.1 Establish develop environment

To install cross compile tool, you should have the Linux distribution pre-installed in your host computer, for example: Ubuntu Linux distribution. Then put the cross compile tool, arm-unknown-linux-gnueabi-4.2.2.tar.bz2, to the "/" folder and use root permission to execute the following command to install it:

```
# tar jxvf arm-unknown-linux-gnueabi-4.2.2.tar.bz2
```

After the installation, the development environment will be established.

You may execute “arm-unknown-linux-gnueabi-gcc -v” to check the development environment is successfully installed if you can see a similar message as below:



```
adv@adv-desktop: ~
File Edit View Terminal Help
adv@adv-desktop:~$ arm-unknown-linux-gnueabi-gcc -v
Using built-in specs.
Target: arm-unknown-linux-gnueabi
Configured with: /tmp/crosstool-ng/targets/src/gcc-4.2.2/configure --build=i686-build_pc-linux-gnu --host=i686-build_pc-linux-gnu --target=arm-unknown-linux-gnueabi --prefix=/usr/local --with-sysroot=/usr/local/arm-unknown-linux-gnueabi//sys-root --enable-languages=c,c++,java --disable-multilib --with-arch=armv5t --with-tune=xscale --with-float=soft --disable-sjlj-exceptions --enable-_cxa_atexit --disable-libmudflap --enable-threads=posix --enable-target-optspace --with-local-prefix=/usr/local/arm-unknown-linux-gnueabi//sys-root --disable-nls --enable-symvers=gnu --enable-c99 --enable-long-long
Thread model: posix
gcc version 4.2.2
adv@adv-desktop:~$
```

Up till now, user can use cross compile to compile your applications. Information about how to develop applications will be introduced in later sections.

Note! *The cross tool chain is tested on following Linux distributions:*



*Fedora: Fedora10 Fedora11 Fedora12 Fedora13 Fedora14 Fedora15
Fedora16*

*Ubuntu: Ubuntu8.04 Ubuntu8.10 Ubuntu9.04 Ubuntu9.10 Ubuntu10.04
Ubuntu10.10 Ubuntu11.04 Ubuntu11.10*

Debian: Debian5.0 Debian6.0

3.2 Quick start to develop application

Here we will give an example to show how to develop an application and transfer your application to the APAX-5522 by using FTP or USB storage device.

The example is HelloWorld.c, the source code of this example is as follows:

```
#include <stdio.h>
int main(void)
{
    printf("Hello World!\n");
    return 0;
}
```

3.2.1 Compile HelloWorld.c

1. Create HelloWorld.c on your Linux PC (for example: Ubuntu)

```
● adv@adv-desktop: ~
File Edit View Terminal Help
#include <stdio.h>
int main(void)
{
    printf("Hello World!\n");
    return 0;
}
```

2. Use “arm UNKNOWN -linux-gnueabi-gcc” command to compile HelloWorld.c to HelloWord

```
● adv@adv-desktop: ~/Documents
File Edit View Terminal Help
adv@adv-desktop:~/Documents$ ls -l
total 8
-rw-r--r-- 1 adv adv 76 2011-12-14 15:09 HelloWorld.c
-rw-r--r-- 1 adv adv 179 2012-03-24 18:18 Makefile
adv@adv-desktop:~/Documents$ arm UNKNOWN -linux-gnueabi-gcc HelloWorld.c -o HelloWord
adv@adv-desktop:~/Documents$ ls -l
total 20
-rwxr-xr-x 1 adv adv 8969 2012-03-24 18:19 HelloWord
-rw-r--r-- 1 adv adv 76 2011-12-14 15:09 HelloWorld.c
-rw-r--r-- 1 adv adv 179 2012-03-24 18:18 Makefile
adv@adv-desktop:~/Documents$
```

3.2.2 Upload “HelloWorld” to the APAX-5522

1. Upload HelloWorld to APAX-5522 by using FTP
Please refer to the above section 3.9 FTP service.
2. Upload HelloWorld to APAX-5522 by using USB storage device.
 - List the partition tables. This USB storage device node is “/dev/sda1”.

172.19.1.105 - PuTTY

```
# fdisk -l

Disk /dev/sda: 250 MB, 250609664 bytes
16 heads, 32 sectors/track, 956 cylinders
Units = cylinders of 512 * 512 = 262144 bytes

   Device Boot   Start     End   Blocks Id System
/dev/sda1   *        1      956  244720   b Win95 FAT32
#
```

- Mount USB storage device.

172.19.1.105 - PuTTY

```
# mount /dev/sda1 /mnt/
# ls /mnt/
HelloWorld
#
```

c. Copy HelloWorld to APAX-5522.

```
172.19.1.105 - PuTTY
# ls
bin dev etc home lib mnt proc sbin tmp usr var
# cp mnt>HelloWorld .
# ls
HelloWorld dev home mnt sbin tmp usr var
bin etc lib proc
#
```

3.2.3 Execute “HelloWorld” on the APAX-5522

1. Change HelloWorld file mode bit as 755.

```
172.19.1.105 - PuTTY
# chmod 777 HelloWorld
# ls -l
-rwxrwxrwx 1 0 0 8970 HelloWorld
drwxr-xr-x 2 1000 1000 0 bin
drwxrwxrwx 4 0 0 0 dev
drwxr-xr-x 9 1000 1000 0 etc
drwxrwxrwx 2 0 0 0 home
drwxr-xr-x 3 1000 1000 0 lib
drwxr-xr-x 2 0 0 2048 mnt
dr-xr-xr-x 41 0 0 0 proc
drwxrwxrwx 2 0 0 0 sbin
drwxrwxrwx 2 0 0 0 tmp
drwxrwxrwx 8 0 0 0 usr
drwxrwxrwx 10 0 0 0 var
#
```

2. Execute HelloWorld program on APAX-5522.

```
172.19.1.105 - PuTTY
# ./HelloWorld
Hello World!
```

3.2.4 Sample Makefile of HelloWorld.c

The following example codes are the sample Makefile of HelloWorld.c.

For example, user can modify the Makefile on demand and put it on your host computer and enter that folder in the host computer, then execute “make” command. The HelloWorld.c will be built.

```
# make
```

Execute “make clean” command will delete the HelloWorld program.

```
# make clean
```

Makefile sample for HelloWorld program

```
CC=arm-unknown-linux-gnueabi-gcc
CFLAGS= -O
BINS=HelloWorld

$(BINS): % : %.c
$(CC) $(CFLAGS) -o $@ $<

clean:
rm -f *.o $(BINS)
```

When compiling, if message similar to “cannot find xxx.h files” appears, please make sure the file is under the search directory. If not, please add the file through “-I path”. Path means the path of the header file.

If “undefined reference to XXX” appears, this means the library of this function may not be linked. Please make sure the library file is under the search directory. If not, please add the file through “-L path”. Path means the path of the library.

Chapter 4

**APAX utility tool kit for
Linux**

The APAX utility tool kit provides eleven instructions that make it convenient for users to access the APAX-5000 I/O modules. They are located in the path: “/bin”. Please see the description as following.

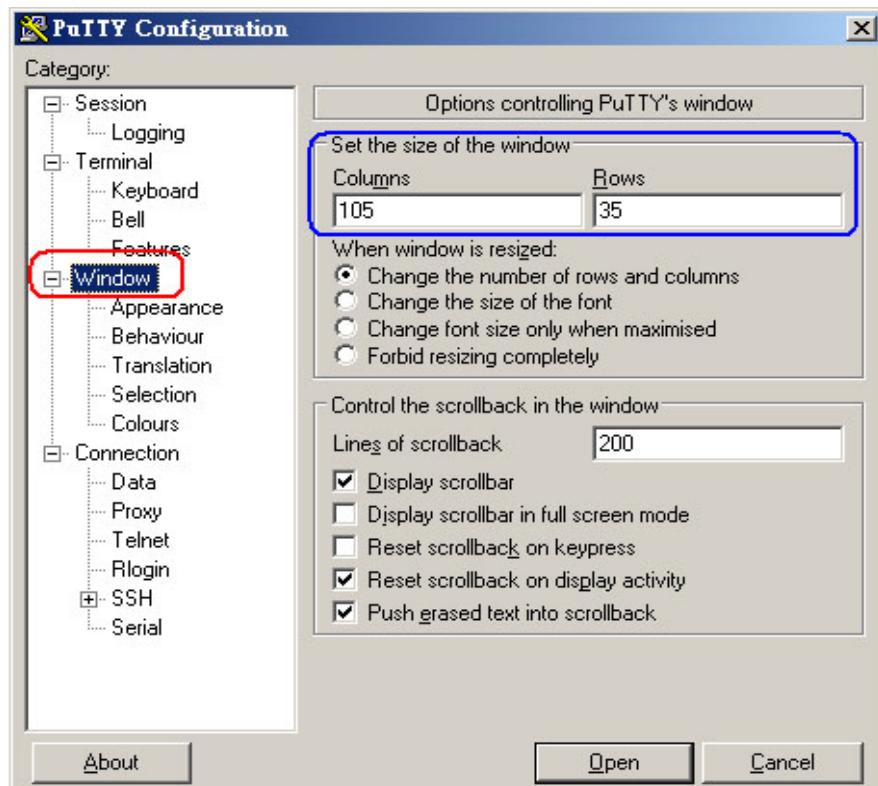
4.1 Overview

Instruction	Function description
apax_UtilityInfo	Display the utility version and support modules
apax_GetModuleList	Discovery the APAX-5000 modules
apax_GetModuleInfo	Get the APAX-5000 modules information
apax_GetChannelInfo	Get the APAX-5000 modules channel information
apax_SetChannelValue	Set the APAX-5000 modules channel output
apax_SetChannelRange	Set the APAX-5000 modules channel range
apax_SetDoConfig	Set the APAX-5000 DO modules configuration
apax_SetAoConfig	Set the APAX-5000 AO modules configuration
apax_SetCntConfig	Set the APAX-5000 CNT modules configuration
apax_TimeStampAI	Get the APAX-5017PE AI module channel with timestamp
apax_TimeStampDI	Get the APAX-5040PE DI module channel with timestamp

Note!



We recommend that when you use command apax_GetChannelInfo which is periodically update function to show the APAX-5000 IO modules channel information, please use Putty or Pierry tool to connect to the APAX-5522 and set the window size of columns and rows as follows for a good display on channel information updating.



- PutTY download page
<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

- PieTTY download page
<http://ntu.csie.org/~piaip/pietty/>

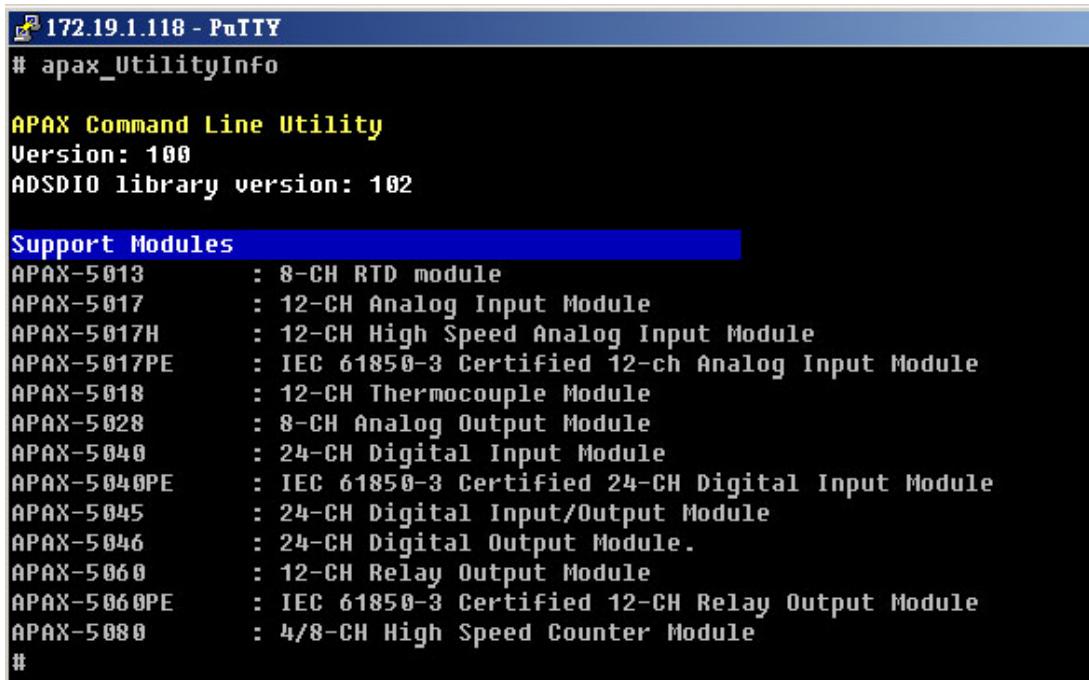
4.2 Command introduction

This section includes important information for users to use APAX utility tool kit. Please refer to the parameters of each command in order to correctly use.

4.2.1 Display APAX utility information

Instruction	Function description
apax.UtilityInfo	Display the utility version and support modules
Usage:	
apax.UtilityInfo	
■ Parameters:	
None.	
■ Result:	
Display the utility version and support modules	

4.2.1.1 Demo



```

172.19.1.118 - PuTTY
# apax.UtilityInfo

APAX Command Line Utility
Version: 100
ADSADIO library version: 102

Support Modules
APAX-5013      : 8-CH RTD module
APAX-5017      : 12-CH Analog Input Module
APAX-5017H     : 12-CH High Speed Analog Input Module
APAX-5017PE    : IEC 61850-3 Certified 12-ch Analog Input Module
APAX-5018      : 12-CH Thermocouple Module
APAX-5028      : 8-CH Analog Output Module
APAX-5040      : 24-CH Digital Input Module
APAX-5040PE    : IEC 61850-3 Certified 24-CH Digital Input Module
APAX-5045      : 24-CH Digital Input/Output Module
APAX-5046      : 24-CH Digital Output Module.
APAX-5060      : 12-CH Relay Output Module
APAX-5060PE    : IEC 61850-3 Certified 12-CH Relay Output Module
APAX-5080      : 4/8-CH High Speed Counter Module
#

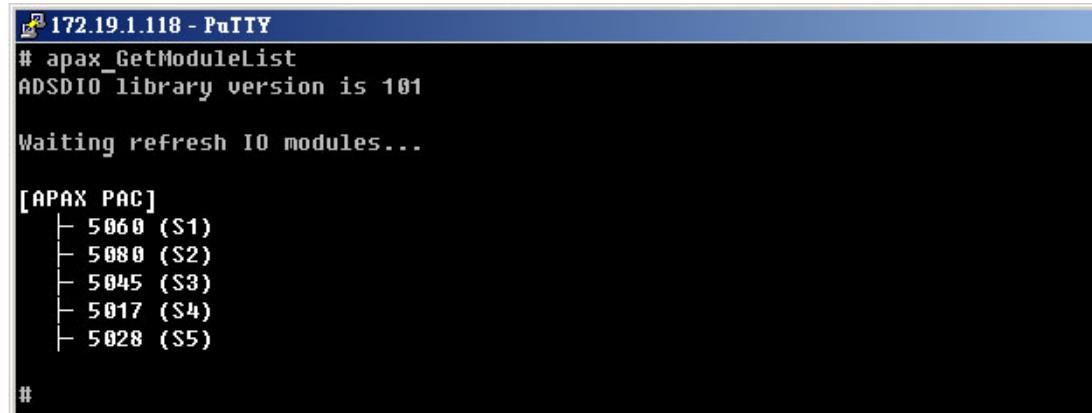
```

4.2.2 Get module list

Instruction	Function description
apax.GetModuleList	Discovery the APAX-5000 modules.
Usage:	
apax.GetModuleList	

-
- Parameters:
None.
 - Result:
Found that the APAX-5000 module name and slot ID.
[APAX PAC]
 └ **Module Name (Slot ID)**
 ...
 ...
-

4.2.2.1 Demo: Discovery the APAX-5000 modules



```

172.19.1.118 - PuTTY
# apax_GetModuleList
ADSDIO library version is 101

Waiting refresh IO modules...

[APAX PAC]
└─ 5060 (S1)
└─ 5080 (S2)
└─ 5045 (S3)
└─ 5017 (S4)
└─ 5028 (S5)

#

```

4.2.3 Get module information

Instruction	Function description
apax_GetModuleInfo	Get the module information of the indicated slot.
Usage:	
	apax_GetModuleInfo SlotID
■ Parameters:	
	None = Display online help
	Slot ID: The slot ID which is ranged from 0 to 31 Slot ID = 255, show all found the APAX-5000 modules.
■ Result:	
	Found that the APAX-5000 module name and slot ID. [Module Name] Switch ID : Description : Firmware : Channel Type :

4.2.3.1 Demo: Get the module information

Online help:

```
172.19.1.118 - PuTTY
# apax_GetModuleInfo
DESCRIPTION:
    Get the module information of the indicated slot
USAGE:
    apax_GetModuleInfo slotID
PARAMETER:
    slotID:
        The slot ID which is ranged from 0 to 31
        slotID = 255 means display all
#
# _
```

Get the one module information:

```
172.19.1.118 - PuTTY
# apax_GetModuleInfo 3
ADSDIO library version is 101

Get Module Information...

[APAX-5045]
Switch ID      :3
Description    :24-CH Digital Input/Output Module
Firmware       :A108
Channel Type   :12 DI channels; 12 DO channels;
#
# _
```

Get all found the module information: SlotID = 255

```

172.19.1.118 - PuTTY
# apax_GetModuleInfo 255
ADSDIO library version is 101

Get Module Information...

[APAX-5060]
Switch ID   :1
Description  :12-CH Relay Output Module
Firmware    :A108
Channel Type:12 DO channels;

[APAX-5080]
Switch ID   :2
Description  :4/8-CH High Speed Counter Module
Firmware    :A110
Channel Type:4 DI channels; 4 DO channels; 8 Counter channels;

[APAX-5045]
Switch ID   :3
Description  :24-CH Digital Input/Output Module
Firmware    :A108
Channel Type:12 DI channels; 12 DO channels;

[APAX-5017]
Switch ID   :4
Description  :12-CH Analog Input Module
Firmware    :A115
AD Version  :A101
Channel Type:12 AI channels;

[APAX-5028]
Switch ID   :5
Description  :8-CH Analog Output Module
Firmware    :A110
Channel Type:8 AO channels;
# -

```

4.2.4 Get channel information

Instruction	Function description
apax_GetChannelInfo	Periodically update the channel information of the indicated slot ID and channel type.

Usage:

apax_GetChannelInfo slotID channelType [-p msec] [-n]

■ Parameters:

None = Display online help

Slot ID:

The slot ID which is ranged from 0 to 31

Channel type:

1 = DI

2 = DO

3 = AI

4 = AO

5 = COUNTER

option:

-p, --Polling interval (msec); default is no delay

-n, --No polling. Get data once

■ Result:

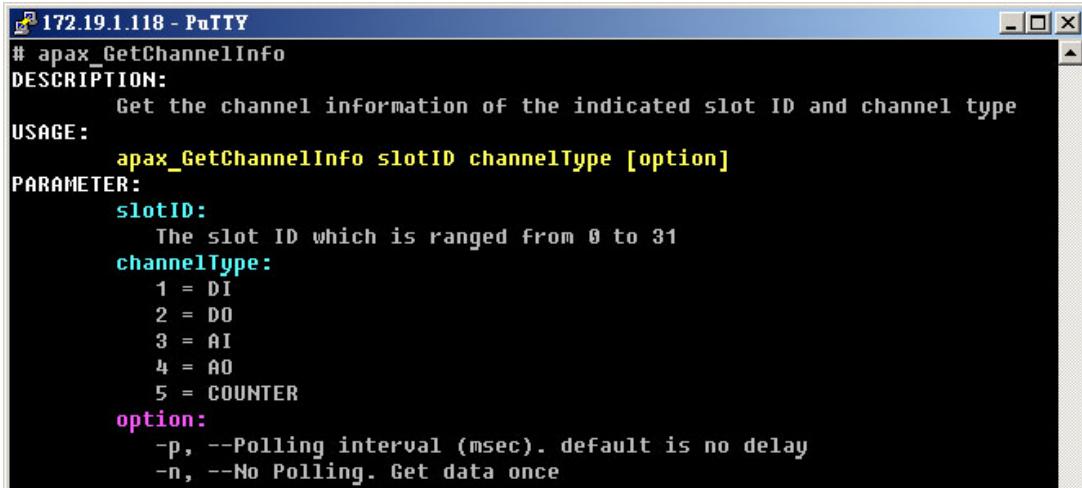
for example AI module

Module Name(Slot ID)

Channel Type	Channel ID	Channel value	Channel status	Channel Range
[Note]				
Press Esc key to exit the program.				

4.2.4.1 Demo: Get DI/DO/AI/AO/CNT channel values

Online help:

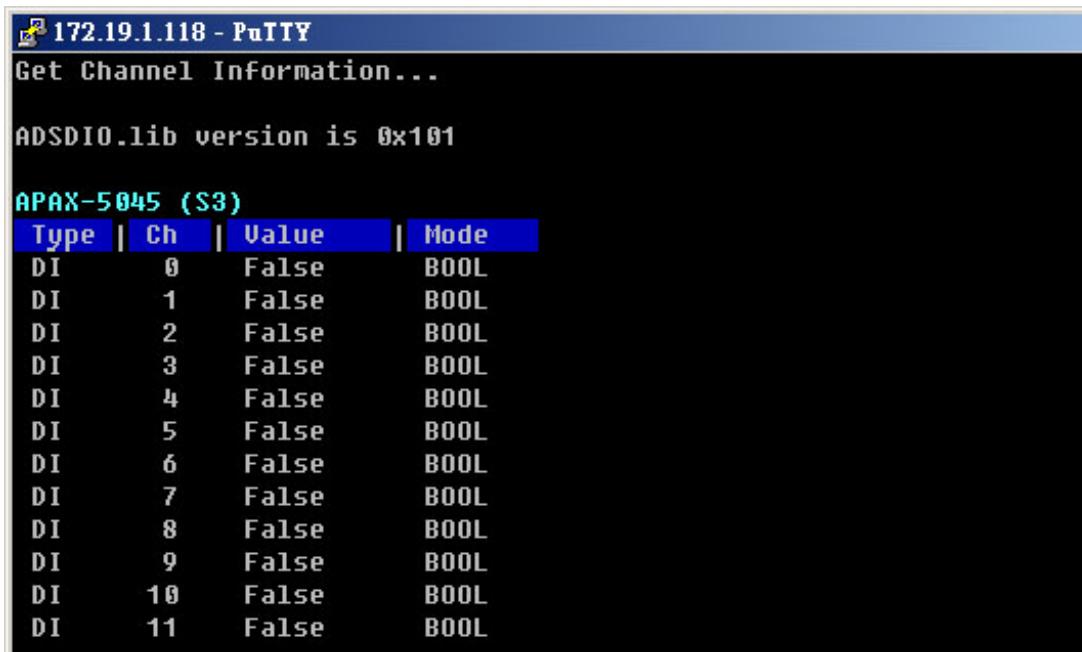


```
# apax_GetChannelInfo
DESCRIPTION:
    Get the channel information of the indicated slot ID and channel type
USAGE:
    apax_GetChannelInfo slotID channelType [option]
PARAMETER:
    slotID:
        The slot ID which is ranged from 0 to 31
    channelType:
        1 = DI
        2 = DO
        3 = AI
        4 = AO
        5 = COUNTER
    option:
        -p, --Polling interval (msec). default is no delay
        -n, --No Polling. Get data once
```

For example: APAX-5045 24-CH Digital Input / Output Module

- Get DI channel info

```
# apax_GetChannelInfo 3 1
```



```
Get Channel Information...
ADSDIO.lib version is 0x101

APAX-5045 (S3)
Type | Ch | Value | Mode
DI   | 0  | False  | BOOL
DI   | 1  | False  | BOOL
DI   | 2  | False  | BOOL
DI   | 3  | False  | BOOL
DI   | 4  | False  | BOOL
DI   | 5  | False  | BOOL
DI   | 6  | False  | BOOL
DI   | 7  | False  | BOOL
DI   | 8  | False  | BOOL
DI   | 9  | False  | BOOL
DI   | 10 | False  | BOOL
DI   | 11 | False  | BOOL
```

■ Get DO channel info

```
# apax_GetChannelInfo 3 2
```

Putty terminal window titled "172.19.1.118 - PuTTY" displaying "Get Channel Information...". The output shows the ADSDIO.lib version is 0x101 and a table for the APAX-5045 module (S3) with 12 digital output channels (DO). All channels are set to False, mode is BOOL, and safety value is Disable.

Type	Ch	Value	Mode	Safety value
DO	0	False	BOOL	Disable
DO	1	False	BOOL	Disable
DO	2	False	BOOL	Disable
DO	3	False	BOOL	Disable
DO	4	False	BOOL	Disable
DO	5	False	BOOL	Disable
DO	6	False	BOOL	Disable
DO	7	False	BOOL	Disable
DO	8	False	BOOL	Disable
DO	9	False	BOOL	Disable
DO	10	False	BOOL	Disable
DO	11	False	BOOL	Disable

For example: APAX-5017 12-CH Analog Input Module

■ Get AI channel info

```
# apax_GetChannelInfo 4 3
```

Putty terminal window titled "172.19.1.118 - PuTTY" displaying "Get Channel Information...". The output shows the ADSDIO.lib version is 0x101 and a table for the APAX-5017 module (S7) with 12 analog input channels (AI). The values range from -0.0001 to 20.0000, and the status for channel 2 is "Burn out".

Type	Ch	Value	Ch Status	Range
AI	0	0.0165	Normal	0~20 mA
AI	1	0.0038	Normal	+/- 10 V
AI	2	20.0000	Burn out	4~20 mA
AI	3	-0.0000	Normal	+/- 1 V
AI	4	-0.0435	Normal	+/- 150 mV
AI	5	-0.0001	Normal	+/- 5 V
AI	6	-0.0343	Normal	+/- 150 mV
AI	7	-0.0002	Normal	+/- 10 V
AI	8	-0.0389	Normal	+/- 150 mV
AI	9	-0.0001	Normal	+/- 5 V
AI	10	20.0000	Burn out	4~20 mA
AI	11	1.5691	Normal	+/- 10 V

For example: APAX-5028 8-CH Analog Output Module

- Get AO channel info

```
# apax_GetChannelInfo 5 4
```

172.19.1.118 - PuTTY

Get Channel Information...

ADSDIO.lib version is 0x101

APAX-5028 (S5)

Type	Ch	Value	Range	Startup	Safety value
AO	0	-0.0000	+/- 2.5 V	-0.0000	Disable
AO	1	-0.0001	+/- 5 V	-0.0001	Disable
AO	2	2.3000	+/- 10 V	2.3000	Disable
AO	3	0.0000	0~20 mA	0.0000	Disable
AO	4	0.0000	0~5 V	0.0000	Disable
AO	5	0.0000	0~10 V	0.0000	Disable
AO	6	4.0000	4~20 mA	4.0000	Disable
AO	7	-0.0002	+/- 10 V	-0.0002	Disable

For example: APAX-5080 4/8-CH High Speed Counter Module

- Get DI channel info

```
# apax_GetChannelInfo 2 1
```

172.19.1.118 - PuTTY

Get Channel Information...

ADSDIO.lib version is 0x101

APAX-5080 (S2)

Type	Ch	Value	Mode
DI	0	False	BOOL
DI	1	False	BOOL
DI	2	False	BOOL
DI	3	False	BOOL

- Get DO channel info

```
# apax_GetChannelInfo 2 2
```

172.19.1.118 - PuTTY

Get Channel Information...

ADSDIO.lib version is 0x101

APAX-5080 (S2)

Type	Ch	Value	Mode	Alarm type	Alarm limit	Alarm flag	Map Ch	DO behavior
Alarm	0	False	BOOL	High	10	False	Cnt0	PulseHi (5000)
Alarm	1	False	BOOL	High	10	False	Cnt2	High level
DO	2	False	BOOL	Disable	Disable	False	Disable	Disable
DO	3	False	BOOL	Disable	Disable	False	Disable	Disable

■ Get CNT channel info

```
# apax_GetChannelInfo 2 5
```

The screenshot shows a PuTTY terminal window titled "172.19.1.118 - PuTTY". The command "Get Channel Information..." was entered, followed by the output:

```
ADSDIO.lib version is 0x101
APAX-5080 (S2)
Type | Ch | Value | Mode | Startup | Counting | Status
CNT  | 0  | 0     | Up and Down mode | 0      | Stop    | Normal
CNT  | 1  | 0     | Up and Down mode | 0      | Stop    | Normal
CNT  | 2  | 0     | Up mode          | 0      | Stop    | Normal
CNT  | 3  | 0     | Frequency mode   | 0      | Stop    | Normal
CNT  | 4  | 0     | AB1X mode        | 0      | Stop    | Normal
CNT  | 5  | 0     | AB1X mode        | 0      | Stop    | Normal
CNT  | 6  | 0     | Up mode          | 0      | Stop    | Normal
CNT  | 7  | 0     | Frequency mode   | 0      | Stop    | Normal
```

4.2.5 Set channel value

Instruction	Function description
apax_SetChannelValue	Set the module output value of the indicated slot.

Set single channel value

Usage:

`apax_GetChannelInfo slotID channelType [-p msec] [-n]`

- Parameters:

None = Display online help

Slot ID:

The slot ID which is ranged from 0 to 31

FuncType:

1 = set single DO value

3 = set single AO value

Channel ID:

The channel ID starts from 0

Value:

The channel value of the indicated channel

[DO module]

1 = High

0 = Low

[AO module]

The channel value specified by the channel range.

- Result:

Show set channel output done or failed message.

Set multi channel values

Usage:

`apax_SetChannelValue slotID funcType DataLen param0 param1 ...`

- Parameters:

None = Display online help

Slot ID:

The slot ID which is ranged from 0 to 31

FuncType:

2 = set multi DO values

4 = set multi AO values

Data length:

The total of the parameter count.

if data length is 3, followed by 3 data (param0 param1 param2)

param0:

[DO module]

The DO values from channel 0 to 31 to be set.

It is DWORD value indicates the channel 0 to 31.

[AO module]

The channels mask. From LSB to MSB of the value indicate.

If the bit is 1, it means that the channel must change value.

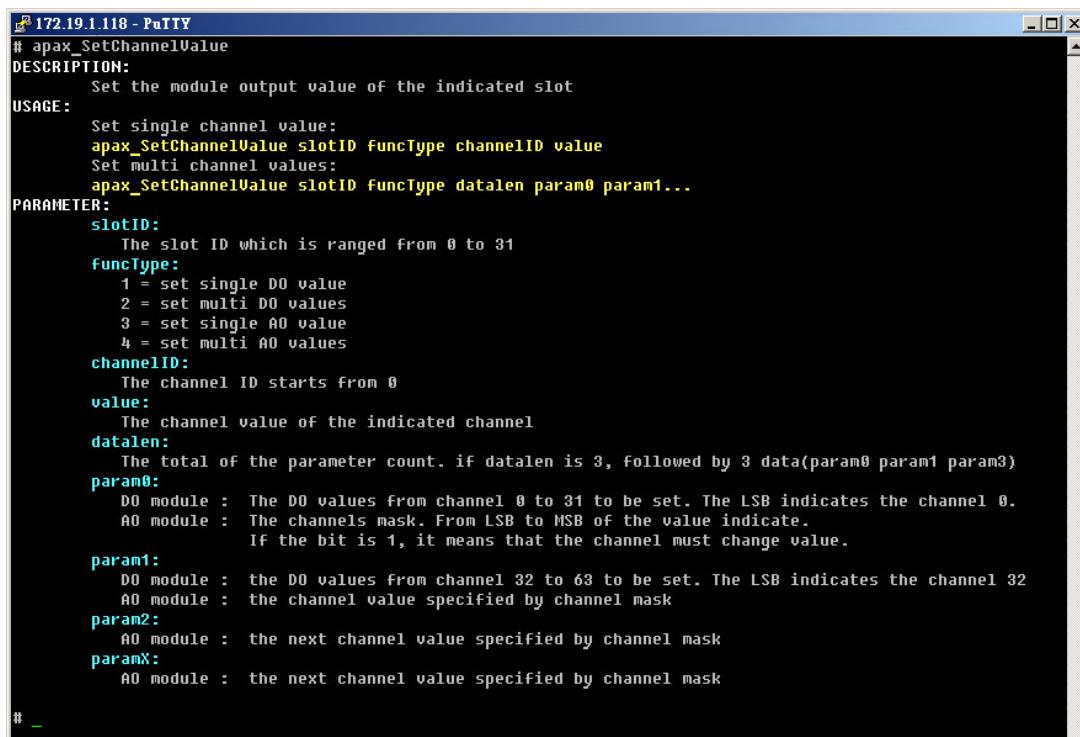
For example: 11110000(0xf0) means the channel 4 to 7 must change value.

- param1:
 - [DO module]
 - The DO values from channel 32 to 63 to be set.
 - It is DWORD value indicates the channel 32 to 63.
 - Bit 0 indicates the channel 32 and bit 31 indicates the channel 63.
 - [AO module]
 - The channel value specified by the channel range.
- param2:
 - [AO module]
 - The next channel value specified by channel range.
- paramX:
 - [AO module]
 - The next channel value specified by channel range.

- Result:
 - Show set channel output done or failed message.

4.2.5.1 Demo: Set DO/AO channel output

Online help:



```
# 172.19.1.118 - PuTTY
# apax_SetChannelValue
DESCRIPTION:
  Set the module output value of the indicated slot
USAGE:
  Set single channel value:
  apax_SetChannelValue slotID funcType channelID value
  Set multi channel values:
  apax_SetChannelValue slotID funcType dataLen param0 param1...
PARAMETER:
  slotID:
    The slot ID which is ranged from 0 to 31
  funcType:
    1 = set single DO value
    2 = set multi DO values
    3 = set single AO value
    4 = set multi AO values
  channelID:
    The channel ID starts from 0
  value:
    The channel value of the indicated channel
  dataLen:
    The total of the parameter count. if dataLen is 3, followed by 3 data(param0 param1 param3)
  param0:
    DO module : The DO values from channel 0 to 31 to be set. The LSB indicates the channel 0.
    AO module : The channels mask. From LSB to MSB of the value indicate.
                If the bit is 1, it means that the channel must change value.
  param1:
    DO module : the DO values from channel 32 to 63 to be set. The LSB indicates the channel 32
    AO module : the channel value specified by channel mask
  param2:
    AO module : the next channel value specified by channel mask
  paramX:
    AO module : the next channel value specified by channel mask

# _
```

For example: APAX-5045 24-CH Digital Input / Output Module

- Set single DO channel output: set channel 0 as HIGH

```
# apax_SetChannelValue 3 1 0 1
```

User can use the apax_GetChannelInfo command to check the result:

APAX-5045 (\$3)				
Type	Ch	Value	Mode	Safety value
DO	0	True	BOOL	Disable
DO	1	False	BOOL	Disable
DO	2	False	BOOL	Disable
DO	3	False	BOOL	Disable
DO	4	False	BOOL	Disable
DO	5	False	BOOL	Disable
DO	6	False	BOOL	Disable
DO	7	False	BOOL	Disable
DO	8	False	BOOL	Disable
DO	9	False	BOOL	Disable
DO	10	False	BOOL	Disable
DO	11	False	BOOL	Disable

- Set multi DO channels output:
Set channel 8 to 11 as HIGH
The DO values will be 111100000000 (0xf00)

```
# apax_SetChannelValue 3 2 1 0xf00
```

User can use the apax_GetChannelInfo command to check the result:

APAX-5045 (\$3)				
Type	Ch	Value	Mode	Safety value
DO	0	False	BOOL	Disable
DO	1	False	BOOL	Disable
DO	2	False	BOOL	Disable
DO	3	False	BOOL	Disable
DO	4	False	BOOL	Disable
DO	5	False	BOOL	Disable
DO	6	False	BOOL	Disable
DO	7	False	BOOL	Disable
DO	8	True	BOOL	Disable
DO	9	True	BOOL	Disable
DO	10	True	BOOL	Disable
DO	11	True	BOOL	Disable

For example: APAX-5028 8-CH Analog Output Module

- Set single AO channel output: set channel 0 output as 2.11 volts

```
# apax_SetChannelValue 5 3 0 2.11
```

User can use the apax_GetChannelInfo command to check the result:

Get Channel Information...						
ADSDIO.lib version is 0x101						
APAX-5028 (S5)						
Type	Ch	Value	Range	Startup	Safety value	
AO	0	2.1100	+/- 2.5 V	-0.0000	Disable	
AO	1	-0.0001	+/- 5 V	-0.0001	Disable	
AO	2	2.3000	+/- 10 V	2.3000	Disable	
AO	3	0.0000	0~20 mA	0.0000	Disable	
AO	4	0.0000	0~5 V	0.0000	Disable	
AO	5	0.0000	0~10 V	0.0000	Disable	
AO	6	4.0000	4~20 mA	4.0000	Disable	
AO	7	-0.0002	+/- 10 V	-0.0002	Disable	

- Set multi AO channels output:
Set channel 3 output as 5.89 mA
Set channel 5 output as 7.25 volts
Set channel 7 output as - 6.475 volts
The channel mask will be 10101000 (0xA8)

```
# apax_SetChannelValue 5 4 4 0xA8 5.89 7.25 -6.475
```

User can use the apax_GetChannelInfo command to check the result:

Get Channel Information...						
ADSDIO.lib version is 0x101						
APAX-5028 (S5)						
Type	Ch	Value	Range	Startup	Safety value	
AO	0	2.1100	+/- 2.5 V	-0.0000	Disable	
AO	1	-0.0001	+/- 5 V	-0.0001	Disable	
AO	2	2.3000	+/- 10 V	2.3000	Disable	
AO	3	5.8900	0~20 mA	0.0000	Disable	
AO	4	0.0000	0~5 V	0.0000	Disable	
AO	5	7.2499	0~10 V	0.0000	Disable	
AO	6	4.0000	4~20 mA	4.0000	Disable	
AO	7	-6.4752	+/- 10 V	-0.0002	Disable	

- Get DO channel info

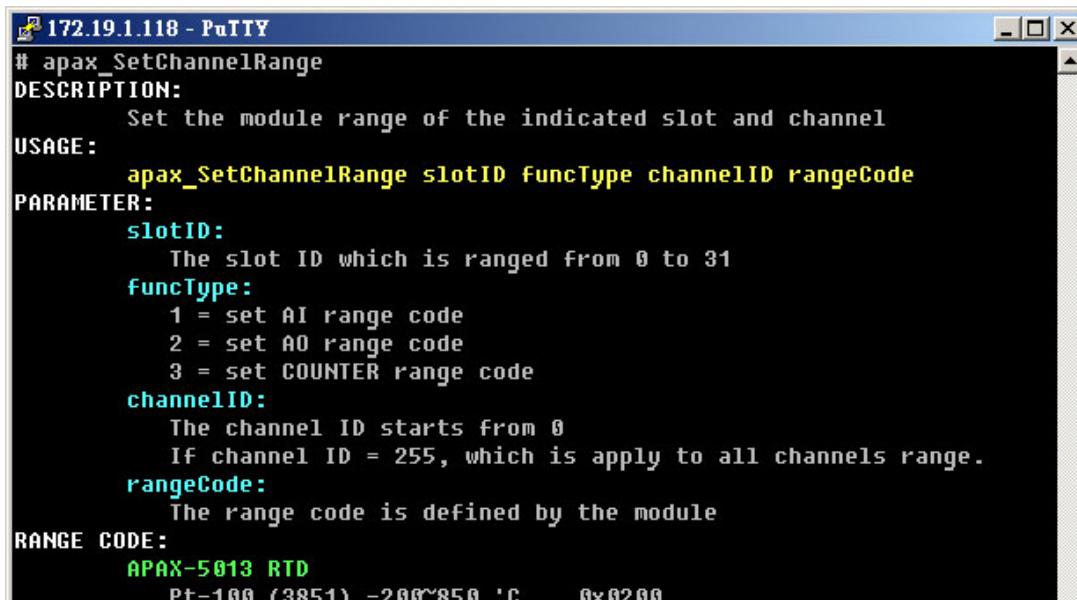
```
# apax_GetChannelInfo 2 2
```

4.2.6 Set channel range

Instruction	Function description
apax_SetChannelRange	Set the channel range of the indicated slot and channel
e	
Usage:	
apax_SetChannelRange slotID funcType channelID rangeCode	
■ Parameters:	
None = Display online help	
Slot ID:	
The slot ID which is ranged from 0 to 31	
FuncType:	
1 = set AI range code	
2 = set AO range code	
3 = set COUNTER range code	
Channel ID:	
The channel ID starts from 0	
If channel ID = 255, which is apply to all channels range.	
Range Code:	
The range code is defined by the module.	
Please refer to the online help or section of appendix A.	
■ Result:	
Show set channel range done or failed message.	

4.2.6.1 Demo: Set AI/AO/CNT channel range

Online help:



The screenshot shows a PuTTY terminal window titled "172.19.1.118 - PuTTY". The command "# apax_SetChannelRange" is entered, followed by the program's usage and parameter descriptions. The output includes the range code for a Pt-100 RTD sensor connected to channel 0 of an APAX-5013 module.

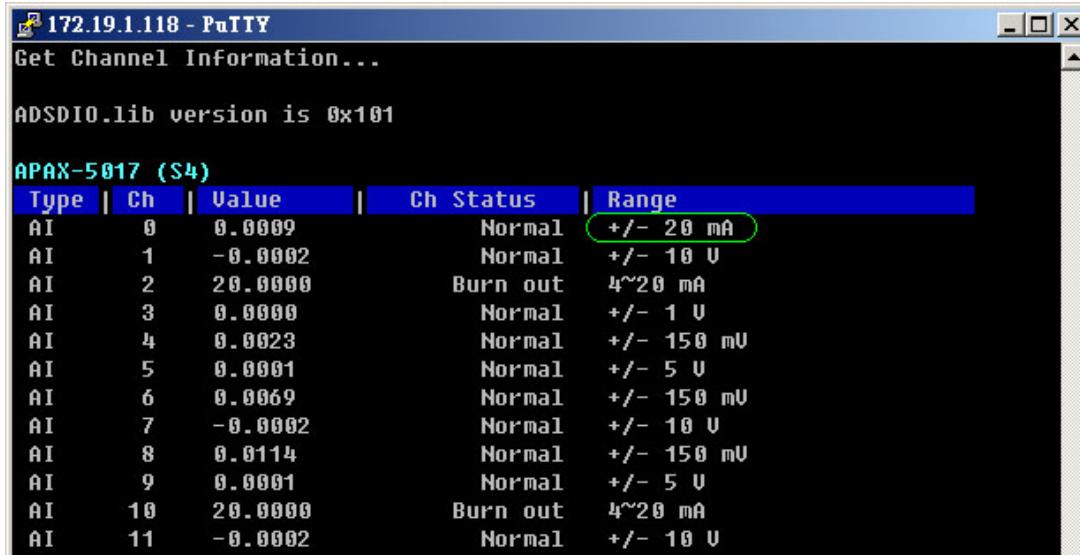
```
# apax_SetChannelRange
DESCRIPTION:
    Set the module range of the indicated slot and channel
USAGE:
    apax_SetChannelRange slotID funcType channelID rangeCode
PARAMETER:
    slotID:
        The slot ID which is ranged from 0 to 31
    funcType:
        1 = set AI range code
        2 = set AO range code
        3 = set COUNTER range code
    channelID:
        The channel ID starts from 0
        If channel ID = 255, which is apply to all channels range.
    rangeCode:
        The range code is defined by the module
RANGE CODE:
    APAX-5013 RTD
    Pt-100 (3851) -200~850 'C      0x0200
```

For example: APAX-5017 12-CH Analog Input Module

- Set AI channel range: set channel 0 range code as 0x0181(+/- 20 mA)

```
# apax_SetChannelRange 4 1 0 0x0181
```

User can use the apax_GetChannelInfo command to check the result:



172.19.1.118 - PuTTY
Get Channel Information...
ADSDIO.lib version is 0x101
APAX-5017 (S4)

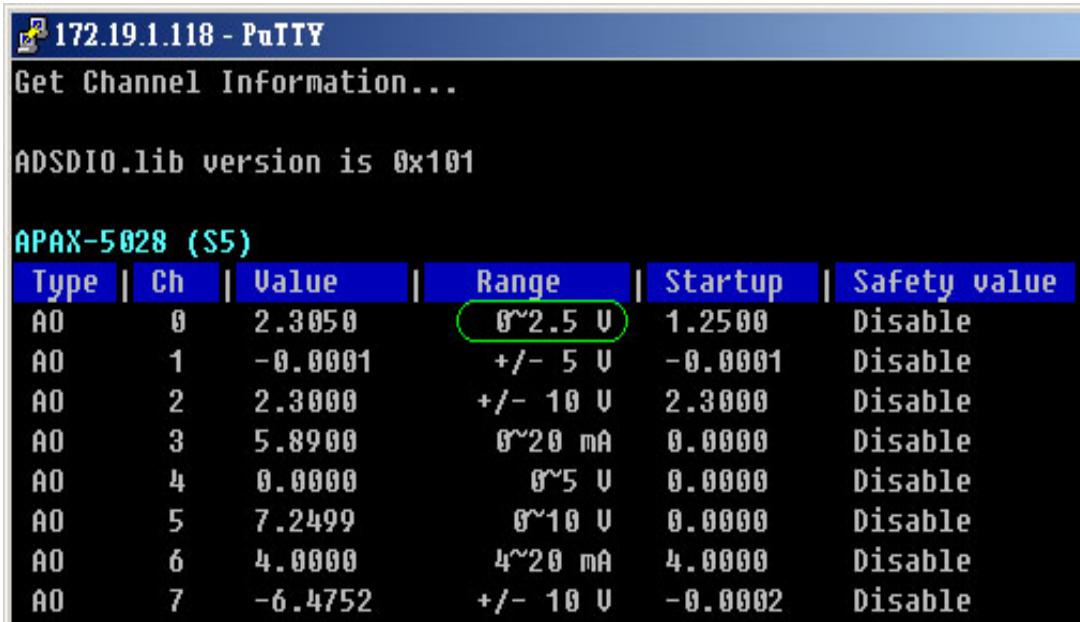
Type	Ch	Value	Ch Status	Range
AI	0	0.0009	Normal	+/- 20 mA
AI	1	-0.0002	Normal	+/- 10 V
AI	2	20.0000	Burn out	4~20 mA
AI	3	0.0000	Normal	+/- 1 V
AI	4	0.0023	Normal	+/- 150 mV
AI	5	0.0001	Normal	+/- 5 V
AI	6	0.0069	Normal	+/- 150 mV
AI	7	-0.0002	Normal	+/- 10 V
AI	8	0.0114	Normal	+/- 150 mV
AI	9	0.0001	Normal	+/- 5 V
AI	10	20.0000	Burn out	4~20 mA
AI	11	-0.0002	Normal	+/- 10 V

For example: APAX-5028 8-CH Analog Output Module

- Set AO channel range: set channel 0 range code as 0x0146(0~2.5 V)

```
# apax_SetChannelRange 5 2 0 0x0146
```

User can use the apax_GetChannelInfo command to check the result:



172.19.1.118 - PuTTY
Get Channel Information...
ADSDIO.lib version is 0x101
APAX-5028 (S5)

Type	Ch	Value	Range	Startup	Safety value
AO	0	2.3050	0~2.5 V	1.2500	Disable
AO	1	-0.0001	+/- 5 V	-0.0001	Disable
AO	2	2.3000	+/- 10 V	2.3000	Disable
AO	3	5.8900	0~20 mA	0.0000	Disable
AO	4	0.0000	0~5 V	0.0000	Disable
AO	5	7.2499	0~10 V	0.0000	Disable
AO	6	4.0000	4~20 mA	4.0000	Disable
AO	7	-6.4752	+/- 10 V	-0.0002	Disable

For example: APAX-5028 4/8-CH High Speed Counter Module

- Set CNT channel range: set channel 7 range code as 0x01c2 (Up mode)

```
# apax_SetChannelRange 2 3 7 0x01c2
```

User can use the apax_GetChannelInfo command to check the result:

```

172.19.1.118 - PuTTY
Get Channel Information...
ADSDIO.lib version is 0x101

APAX-5080 (S2)
Type | Ch | Value | Mode | Startup | Counting | Status
CNT  | 0  | 0     | Up and Down mode | 0       | Stop    | Normal
CNT  | 1  | 0     | Up and Down mode | 0       | Stop    | Normal
CNT  | 2  | 0     | Up mode          | 0       | Stop    | Normal
CNT  | 3  | 0     | Frequency mode   | 0       | Stop    | Normal
CNT  | 4  | 0     | AB1X mode        | 0       | Stop    | Normal
CNT  | 5  | 0     | AB1X mode        | 0       | Stop    | Normal
CNT  | 6  | 0     | Up mode          | 0       | Stop    | Normal
CNT  | 7  | 0     | Up mode          | 0       | Stop    | Normal

```

For example: APAX-5017 12-CH Analog Input Module

- Set all channel range code as 0x0143 (+/- 10 V)

```
# apax_SetChannelRange 4 1 255 0x0143
```

User can use the apax_GetChannelInfo command to check the result:

```

172.19.1.118 - PuTTY
Get Channel Information...
ADSDIO.lib version is 0x101

APAX-5017 (S4)
Type | Ch | Value | Ch Status | Range
AI   | 0  | -0.0002 | Normal    | +/- 10 V
AI   | 1  | -0.0002 | Normal    | +/- 10 V
AI   | 2  | -0.0002 | Normal    | +/- 10 V
AI   | 3  | -0.0002 | Normal    | +/- 10 V
AI   | 4  | -0.0002 | Normal    | +/- 10 V
AI   | 5  | -0.0002 | Normal    | +/- 10 V
AI   | 6  | -0.0002 | Normal    | +/- 10 V
AI   | 7  | -0.0002 | Normal    | +/- 10 V
AI   | 8  | -0.0002 | Normal    | +/- 10 V
AI   | 9  | -0.0002 | Normal    | +/- 10 V
AI   | 10 | -0.0002 | Normal    | +/- 10 V
AI   | 11 | -0.0002 | Normal    | +/- 10 V

```

4.2.7 Set DO module configuration

Instruction	Function description
apax_SetDoConfig	Set the DO module configuration of the indicated slot
Usage:	
	apax_SetDoConfig slotID funcCode param0 [param1]

- Parameters:
 - None = Display online help
 - Slot ID:
 - The slot ID which is ranged from 0 to 31
 - FuncCode:
 - 1 = set Safety function
 - 2 = set Safety values
 - Parm0:
 - funcCode = 1: Enable: 1 / Disable: 0
 - funcCode = 2: The safety values from channel 0 to 31 to be set.
The LSB indicates the channel 0.
 - Param1:
 - funcCode = 1: Not use
 - funcCode = 2: The safety values from channel 32 to 63 to be set
The LSB indicates the channel 32
- Result:
 - Show set DO configuration done or failed message.

4.2.7.1 Demo: Set DO configuration

Online help:

```
172.19.1.118 - PuTTY
# apax_SetDoConfig
DESCRIPTION:
    Set the DO module configuration of the indicated slot
USAGE:
    apax_SetDoConfig slotID funcCode param0 [param1]
PARAMETER:
    slotID:
        The slot ID which is ranged from 0 to 31
    funcCode:
        1 = set Safety function
        2 = set Safety values
    param0:
        funcCode = 1: Enable: 1 / Disable: 0
        funcCode = 2: The safety values from channel 0 to 31 to be set.
                    The LSB indicates the channel 0.
    param1:
        funcCode = 1: Not use
        funcCode = 2: The safety values from channel 32 to 63 to be set
                    The LSB indicates the channel 32
```

For example: APAX-5060 12-CH Relay Output Module

- Set safety function enable

```
# apax_SetDoConfig 1 1 1
```

For example: APAX-5028 4/8-CH High Speed Counter Module

- Set channel 8 to 11 safety value as HIGH

```
# apax_SetDoConfig 1 2 0xf00
```

User can use the apax_GetChannellInfo command to check the result:

Before:

```
172.19.1.118 - PuTTY
Get Channel Information...
ADSDIO.lib version is 0x101

APAX-5060 (S1)
Type | Ch | Value | Mode | Safety value
DO   | 0  | False  | BOOL | Disable
DO   | 1  | False  | BOOL | Disable
DO   | 2  | False  | BOOL | Disable
DO   | 3  | False  | BOOL | Disable
DO   | 4  | False  | BOOL | Disable
DO   | 5  | False  | BOOL | Disable
DO   | 6  | False  | BOOL | Disable
DO   | 7  | False  | BOOL | Disable
DO   | 8  | False  | BOOL | Disable
DO   | 9  | False  | BOOL | Disable
DO   | 10 | False  | BOOL | Disable
DO   | 11 | False  | BOOL | Disable
```

After:

```
172.19.1.118 - PuTTY
Get Channel Information...
ADSDIO.lib version is 0x101

APAX-5060 (S1)
Type | Ch | Value | Mode | Safety value
DO   | 0  | False  | BOOL | False
DO   | 1  | False  | BOOL | False
DO   | 2  | False  | BOOL | False
DO   | 3  | False  | BOOL | False
DO   | 4  | False  | BOOL | False
DO   | 5  | False  | BOOL | False
DO   | 6  | False  | BOOL | False
DO   | 7  | False  | BOOL | False
DO   | 8  | False  | BOOL | True
DO   | 9  | False  | BOOL | True
DO   | 10 | False  | BOOL | True
DO   | 11 | False  | BOOL | True
```

4.2.8 Set AO module configuration

Instruction	Function description
apax_SetAoConfig	Set the AO module configuration of the indicated slot
Usage:	
apax_SetAoConfig slotID funcCode param0 [param1]	

-
- Parameters:
 - None = Display online help
 - Slot ID:
 - The slot ID which is ranged from 0 to 31
 - FuncCode:
 - 1 = set Safety function
 - 2 = set Safety values
 - 3 = set Startup values
 - Parm0:
 - funcCode = 1: Enable: 1 / Disable: 0
 - funcCode = 2, 3: param0 is channel ID. The channel ID start from 0
If channel ID = 255, which is apply to all channels value.
 - Param1:
 - funcCode = 1: Not use
 - funcCode = 2: The value for the indicated channel
 - Result:
 - Show set AO configuration done or failed message.

4.2.8.1 Demo: Set AO configuration

Online help:

```
172.19.1.118 - PuTTY
# apax_SetAoConfig
DESCRIPTION:
  Set the AO module configuration of the indicated slot
USAGE:
  apax_SetAoConfig slotID FuncCode param0 [param1]
PARAMETER:
  slotID:
    The slot ID which is ranged from 0 to 31
  FuncCode:
    1 = set Safety function
    2 = set Safety values
    3 = set Startup values
  param0:
    FuncCode = 1: Enable: 1 / Disable: 0
    FuncCode = 2, 3: param0 is channel ID. The channel ID start from 0
                  If channel ID = 255, which is apply to all channels value.
  param1:
    FuncCode = 1: Not use
    FuncCode = 2, 3: The value for the indicated channel
```

For example: APAX-5028 8-CH Analog Output Module

- Set safety function enable

```
# apax_SetAoConfig 5 1 1
```

- Set channel 0 safety value as 2.3 volts

```
# apax_SetAoConfig 5 2 0 2.30
```

- Set channel 4 startup value as 2.5 volts

```
# apax_SetAoConfig 5 3 4 2.5
```

User can use the apax_GetChannelInfo command to check the result:

Before:

```
172.19.1.118 - PuTTY
Get Channel Information...
ADSDIO.lib version is 0x101

APAX-5028 (S5)
Type | Ch | Value | Range | Startup | Safety value
AO   | 0  | 2.3050 | 0~2.5 V | 0.0000 | Disable
AO   | 1  | -0.0001 | +/- 5 V | -0.0001 | Disable
AO   | 2  | 2.3000 | +/- 10 V | -0.0002 | Disable
AO   | 3  | 5.8900 | 0~20 mA | 0.0000 | Disable
AO   | 4  | 0.0000 | 0~5 V  | 0.0000 | Disable
AO   | 5  | 7.2499 | 0~10 V | 0.0000 | Disable
AO   | 6  | 4.0000 | 4~20 mA | 4.0000 | Disable
AO   | 7  | -6.4752 | +/- 10 V | -0.0002 | Disable
```

After:

```
172.19.1.118 - PuTTY
Get Channel Information...
ADSDIO.lib version is 0x101

APAX-5028 (S5)
Type | Ch | Value | Range | Startup | Safety value
AO   | 0  | 2.3050 | 0~2.5 V | 0.0000 | 2.3000
AO   | 1  | -0.0001 | +/- 5 V | -0.0001 | -0.0001
AO   | 2  | 2.3000 | +/- 10 V | -0.0002 | -0.0002
AO   | 3  | 5.8900 | 0~20 mA | 0.0000 | 0.0000
AO   | 4  | 0.0000 | 0~5 V  | 2.5000 | 0.0000
AO   | 5  | 7.2499 | 0~10 V | 0.0000 | 0.0000
AO   | 6  | 4.0000 | 4~20 mA | 4.0000 | 4.0000
AO   | 7  | -6.4752 | +/- 10 V | -0.0002 | -0.0002
```

4.2.9 Set CNT module configuration

Instruction	Function description
apax_SetCntConfig	Set the CNT module configuration of the indicated slot
Usage:	
apax_SetCntConfig slotID funcCode channelID param0 [param1]...	

-
- Parameters:
 - None = Display online help
 - Slot ID:
 - The slot ID which is ranged from 0 to 31
 - FuncCode:
 - 1 = set Counting Start/Stop
 - 2 = Reset counting
 - 3 = set Startup values
 - 4 = Clear status
 - 5 = Alarm config
 - 6 = Alarm latch clear
 - Channel ID:
 - The channel ID starts from 0
 - If channel ID = 255, which is apply to all channels.
 - Parm0:
 - funcCode = 1, Start: 1 / Stop: 0
 - funcCode = 2, Not use
 - funcCode = 3, The startup value for the indicated channel
 - funcCode = 4, Not use
 - funcCode = 5, Alarm mode (Enable: 1 / Disable: 0)
 - funcCode = 6, Not use
 - Param1:
 - funcCode = 5, Alarm type (High alarm: 1 / Low alarm: 0)
 - Param2:
 - funcCode = 5, Limit value. The counter limit which will fire the alarm.
 - Param3:
 - funcCode = 5, MapChannel. The counter channel that the alarm mapped to.
 - Param4:
 - funcCode = 5, DO output mode
 - 0 - Low level
 - 1 - High level
 - 2 - Low pulse
 - 3 - High pulse
 - Param5:
 - funcCode = 5, DO pulse width (milliseconds)
 - Result:
 - Show set CNT configuration done or failed message.

4.2.9.1 Demo: Set CNT configuration

Online help:

```

172.19.1.118 - PuTTY
# apax_SetCntConfig
DESCRIPTION:
    Set the COUNT module configuration of the indicated slot
USAGE:
    apax_SetCntConfig slotID funcCode channelID param0 [param1] [param2]...
PARAMETER:
    slotID:
        The slot ID which is ranged from 0 to 31
    funcCode:
        1 = set Counting Start/Stop
        2 = Reset counting
        3 = set Startup values
        4 = Clear status
        5 = Alarm config
        6 = Alarm latch clear
    channelID:
        The channel ID starts from 0
        If channel ID = 255, which is apply to all channels.
    param0:
        FuncCode = 1, Start: 1 / Stop: 0
        FuncCode = 2, Not use
        FuncCode = 3, The startup value for the indicated channel
        FuncCode = 4, Not use
        FuncCode = 5, Alarm mode (Enable: 1 / Disable: 0)
        FuncCode = 6, Not use
    param1:
        FuncCode = 5, Alarm type (High alarm: 1 / Low alarm: 0)
    param2:
        FuncCode = 5, Limit value. The counter limit which will fire the alarm.
    param3:
        FuncCode = 5, MapChannel. The counter channel that the alarm mapped to.
    param4:
        FuncCode = 5, DO output mode
            0 - Low level
            1 - High level
            2 - Low pulse
            3 - High pulse
    param5:
        FuncCode = 5, DO pulse width (milliseconds)

```

For example: APAX-5028 4/8-CH High Speed Counter Module

- Set channel 0 startup value as 10. -----(1)

```
# apax_SetCntConfig 2 3 0 10
```

- Reset channel 0 counter value. -----(2)

```
# apax_SetCntConfig 2 2 0
```

- Set all channel counting start. -----(3)

```
# apax_SetCntConfig 2 1 255 1
```

User can use the apax_GetChannelInfo command to check the result:

Before:

APAX-5080 (S2)							
Type	Ch	Value	Mode	Startup	Counting	Status	
CNT	0	0	Up and Down mode	0	Stop	Normal	
CNT	1	0	Up and Down mode	0	Stop	Normal	
CNT	2	0	Up mode	0	Stop	Normal	
CNT	3	0	Frequency mode	0	Stop	Normal	
CNT	4	0	AB1X mode	0	Stop	Normal	
CNT	5	0	AB1X mode	0	Stop	Normal	
CNT	6	0	Up mode	0	Stop	Normal	
CNT	7	0	Up mode	0	Stop	Normal	

After:

APAX-5080 (S2)							
Type	Ch	Value	Mode	Startup	Counting	Status	
CNT	0	10	Up and Down mode	10	Start	Normal	
CNT	1	10	Up and Down mode	10	Start	Normal	
CNT	2	0	Up mode	0	Start	Normal	
CNT	3	0	Frequency mode	0	Start	Normal	
CNT	4	0	AB1X mode	0	Start	Normal	
CNT	5	0	AB1X mode	0	Start	Normal	
CNT	6	0	Up mode	0	Start	Normal	
CNT	7	0	Up mode	0	Start	Normal	

If the count status abnormal. The counting will stop. Users can clear the status as following command:

- Clear channel 0 count status.

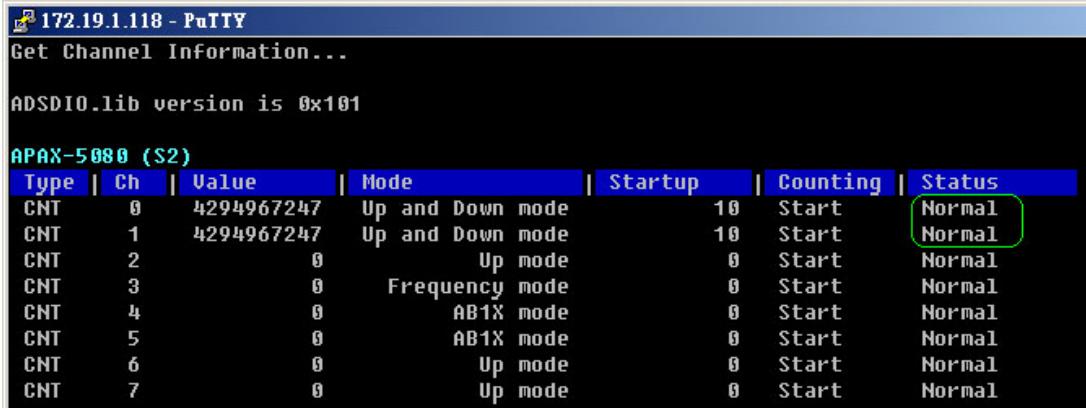
```
# apax_SetCntConfig 2 4 0
```

User can use the apax_GetChannelInfo command to check the result:

Before:

APAX-5080 (S2)							
Type	Ch	Value	Mode	Startup	Counting	Status	
CNT	0	4294967295	Up and Down mode	10	Start	Underflow	
CNT	1	4294967295	Up and Down mode	10	Start	Underflow	
CNT	2	0	Up mode	0	Start	Normal	
CNT	3	0	Frequency mode	0	Start	Normal	
CNT	4	0	AB1X mode	0	Start	Normal	
CNT	5	0	AB1X mode	0	Start	Normal	
CNT	6	0	Up mode	0	Start	Normal	
CNT	7	0	Up mode	0	Start	Normal	

After:



```

172.19.1.118 - PuTTY
Get Channel Information...
ADSDIO.lib version is 0x101

APAX-5028 (S2)
Type | Ch | Value | Mode | Startup | Counting | Status
CNT  | 0  | 4294967247 | Up and Down mode | 10 | Start | Normal
CNT  | 1  | 4294967247 | Up and Down mode | 10 | Start | Normal
CNT  | 2  | 0          | Up mode      | 0   | Start | Normal
CNT  | 3  | 0          | Frequency mode | 0   | Start | Normal
CNT  | 4  | 0          | AB1X mode    | 0   | Start | Normal
CNT  | 5  | 0          | AB1X mode    | 0   | Start | Normal
CNT  | 6  | 0          | Up mode      | 0   | Start | Normal
CNT  | 7  | 0          | Up mode      | 0   | Start | Normal

```

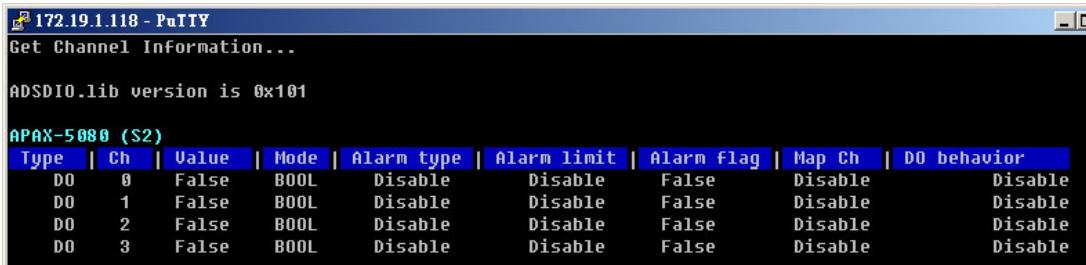
The APAX-5028 DO channel can be configured as an alarm channel. When the count value exceeds the limit, the alarm will occur.

- Set DO channel 0 as an alarm channel. You can use below command to enable alarm mode.

```
# apax_SetCntConfig 2 5 0 1
```

User can use the apax_GetChannelInfo command to check the result:

Before:



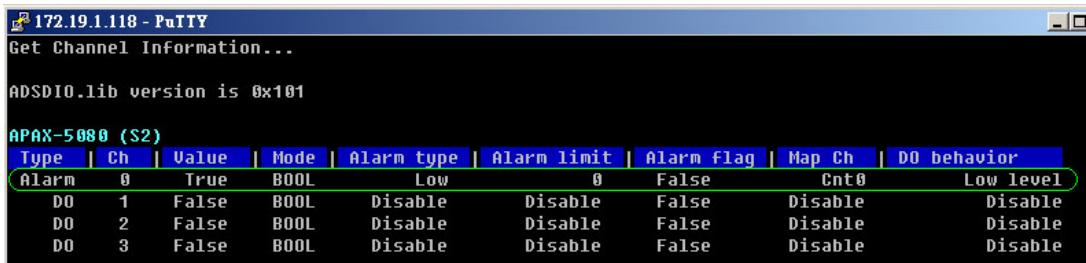
```

172.19.1.118 - PuTTY
Get Channel Information...
ADSDIO.lib version is 0x101

APAX-5028 (S2)
Type | Ch | Value | Mode | Alarm type | Alarm limit | Alarm flag | Map Ch | DO behavior
DO   | 0  | False  | BOOL  | Disable    | Disable    | False     | Disable  | Disable
DO   | 1  | False  | BOOL  | Disable    | Disable    | False     | Disable  | Disable
DO   | 2  | False  | BOOL  | Disable    | Disable    | False     | Disable  | Disable
DO   | 3  | False  | BOOL  | Disable    | Disable    | False     | Disable  | Disable

```

After:



```

172.19.1.118 - PuTTY
Get Channel Information...
ADSDIO.lib version is 0x101

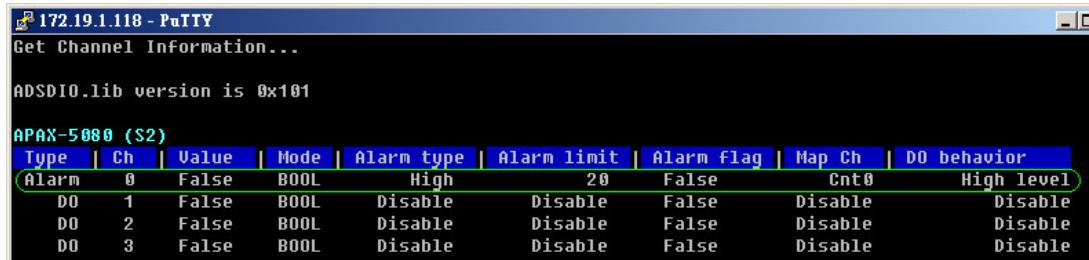
APAX-5028 (S2)
Type | Ch | Value | Mode | Alarm type | Alarm limit | Alarm flag | Map Ch | DO behavior
Alarm | 0  | True   | BOOL  | Low       | 0          | False     | Cnt0   | Low level
DO   | 1  | False  | BOOL  | Disable   | Disable    | False     | Disable | Disable
DO   | 2  | False  | BOOL  | Disable   | Disable    | False     | Disable | Disable
DO   | 3  | False  | BOOL  | Disable   | Disable    | False     | Disable | Disable

```

- Set the channel 0 alarm configuration.
Set Alarm type as High trigger.
Set Alarm limit as 20.
Set Alarm mapping channel as CNT channel 0.
Set DO output behavior as high level.

```
# apax_SetCntConfig 2 5 0 1 1 20 0 1
```

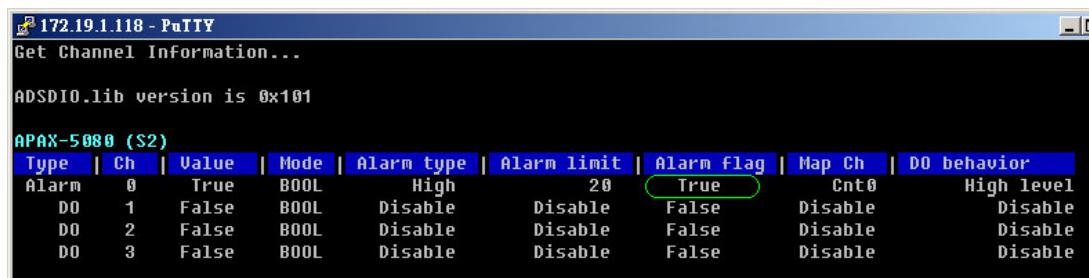
User can use the apax_GetChannelInfo command to check the result:



```
172.19.1.118 - PuTTY
Get Channel Information...
ADSDIO.lib version is 0x101

APAX-5080 (S2)
Type | Ch | Value | Mode | Alarm type | Alarm limit | Alarm flag | Map Ch | DO behavior
Alarm 0 False BOOL High 20 False Cnt0 High level
DO 1 False BOOL Disable Disable False Disable Disable
DO 2 False BOOL Disable Disable False Disable Disable
DO 3 False BOOL Disable Disable False Disable Disable
```

When the count value exceeds the limit, the alarm will occur and the DO channel 0 will remain at high level until the alarm flag is cleared to 0.

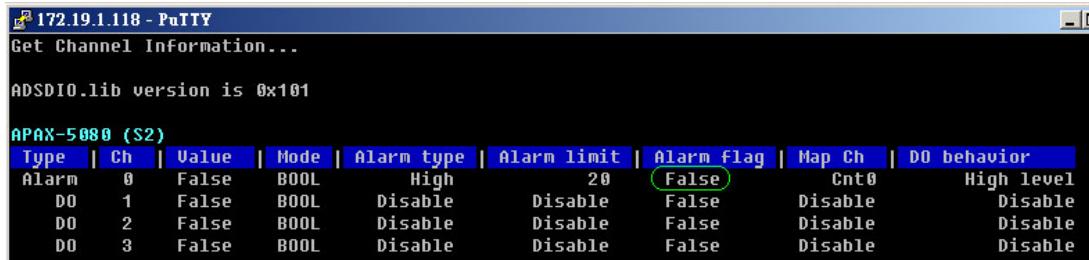


```
172.19.1.118 - PuTTY
Get Channel Information...
ADSDIO.lib version is 0x101

APAX-5080 (S2)
Type | Ch | Value | Mode | Alarm type | Alarm limit | Alarm flag | Map Ch | DO behavior
Alarm 0 True BOOL High 20 True Cnt0 High level
DO 1 False BOOL Disable Disable False Disable Disable
DO 2 False BOOL Disable Disable False Disable Disable
DO 3 False BOOL Disable Disable False Disable Disable
```

- Clear channel 0 alarm flag

```
# apax_SetCntConfig 2 6 0
```



```
172.19.1.118 - PuTTY
Get Channel Information...
ADSDIO.lib version is 0x101

APAX-5080 (S2)
Type | Ch | Value | Mode | Alarm type | Alarm limit | Alarm flag | Map Ch | DO behavior
Alarm 0 False BOOL High 20 False Cnt0 High level
DO 1 False BOOL Disable Disable False Disable Disable
DO 2 False BOOL Disable Disable False Disable Disable
DO 3 False BOOL Disable Disable False Disable Disable
```

4.2.10 Get the APAX-5017PE AI module with timestamp

Instruction	Function description
apax_TimeStampAI	Get the APAX-5017PE AI channel values with timestamp
Usage:	
apax_TimeStampAI slotID param	

-
- Parameters:
 - Slot ID:
 - The slot ID which is ranged from 0 to 31
 - Param:
 - Polling interval (sec). default is 60 sec.
 - If param is 0 means infinite
 - Result:
 - Show the AI channel value with timestamp.
 - Single AI Data [Range code]
 - Channel index
 - Time offset (ms)

4.2.10.1 Demo

Online help:

```
172.19.2.108 - PuTTY
# apax_TimeStampAI
DESCRIPTION:
    Get the APAX-5017PE AI channel values with timestamp of the indicated slot
USAGE:
    apax_TimeStampAI slotID param
PARAMETER:
    slotID:
        The slot ID which is ranged from 0 to 15
    param:
        Polling interval (sec). default is 60 sec.
        If param is 0 means infinite
```

For example: Get APAX-5017PE AI channel data within 1 second. (Slot id is 1)

```
# apax_TimeStampAI 1 1
```

```
172.19.2.108 - PuTTY
# apax_TimeStampAI 1 1
==== APAX-5017PE AI Timestamp example ====
ADSDIO library version is 102

Timestamp start from time:
Thu Jan  1 01:24:46 2012
=====
Get 1 data
Single AI Data = -0.9837 [+/- 1 V]
Channel No. = 9
(offset) = 27 ms

Get 1 data
Single AI Data = 0.0513 [0~20 mA]
Channel No. = 10
(offset) = 110 ms

Get 1 data
Single AI Data = 16.3729 [0~20 mA]
Channel No. = 11
(offset) = 193 ms

Get 1 data
Single AI Data = 10.0000 [+/- 10 V]
Channel No. = 0
(offset) = 277 ms

~//~ Omission

Get 1 data
Single AI Data = 0.0449 [+/- 5 V]
Channel No. = 5
(offset) = 694 ms

Get 1 data
Single AI Data = 0.0266 [+/- 5 V]
Channel No. = 6
(offset) = 777 ms

Get 1 data
Single AI Data = 5.0000 [+/- 5 V]
Channel No. = 7
(offset) = 861 ms

Get 1 data
Single AI Data = 12.0338 [4~20 mA]
Channel No. = 8
(offset) = 944 ms

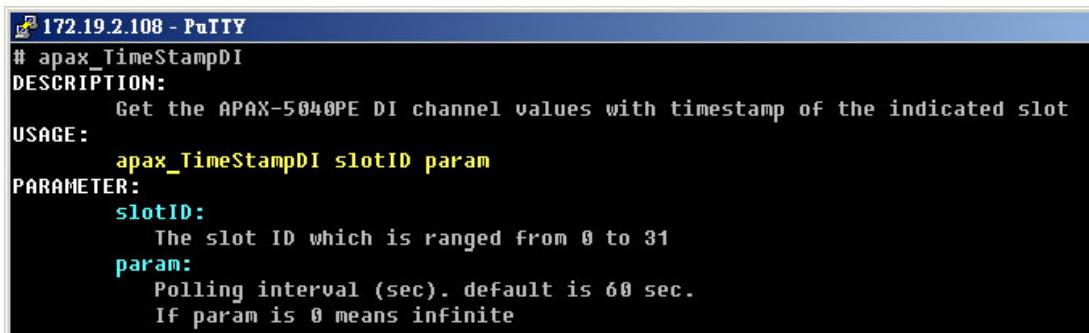
Total 12 data got within 1 sec.
/**/END/**/
```

4.2.11 Get the APAX-5040PE DI module with timestamp

Instruction	Function description
apax_TimeStampDI	Get the APAX-5040PE DI channel values with timestamp
Usage:	
	apax_TimeStampDI slotID param
■ Parameters:	
Slot ID:	The slot ID which is ranged from 0 to 31
Param:	Polling interval (sec). default is 60 sec. If param is 0 means infinite
■ Result:	Show the DI channel values with timestamp. DI Hex Data (32 bits): The LSB is channel 0, MSB is channel 31 Time offset (ms)

4.2.11.1 Demo

Online help:



```
172.19.2.108 - PuTTY
# apax_TimeStampDI
DESCRIPTION:
    Get the APAX-5040PE DI channel values with timestamp of the indicated slot
USAGE:
    apax_TimeStampDI slotID param
PARAMETER:
    slotID:
        The slot ID which is ranged from 0 to 31
    param:
        Polling interval (sec). default is 60 sec.
        If param is 0 means infinite
```

For example: Get APAX-5040PE DI channel data within 5 second. (Slot id is 0)

```
# apax_TimeStampDI 0 5
```

```
172.19.2.108 - PuTTY
# apax_TimeStampDI 0 5
===== APAX-5040PE DI Timestamp example =====
ADSDIO library version is 102

Timestamp start from time:
Fri Jul 13 12:02:07 2012
=====
Get 1 data
DI 32 Data = 0x1
(offset) = 2020 ms
-----
Get 6 data
DI 32 Data = 0x3
(offset) = 2021 ms
-----
DI 32 Data = 0x7
(offset) = 2022 ms
-----
DI 32 Data = 0xF
(offset) = 2023 ms
-----
~//~ Omission
-----
DI 32 Data = 0x7
(offset) = 4022 ms
-----
DI 32 Data = 0xF
(offset) = 4023 ms
-----
DI 32 Data = 0x1F
(offset) = 4024 ms
-----
DI 32 Data = 0x3F
(offset) = 4025 ms
-----
DI 32 Data = 0x7F
(offset) = 4026 ms
-----
Get 1 data
DI 32 Data = 0x60
(offset) = 4520 ms
-----
Get 1 data
DI 32 Data = 0x0
(offset) = 4520 ms
-----
Total 27 data got within 5 sec.
/** END ***/
```

The APAX-5040PE has 24 DI channels. If the DI 32 Data is 0x01 that means channel 0 is high level and the rest of channel 1 to 23 is low level.

Appendix A

**APAX 5000 I/O module
range code settings**

A.1 ADAM/APAX .NET Utility General Window

These ranges are provided for reference. Not all boards support all ranges. Please see hardware manual for valid ranges for a particular board.

	Setting Type	Value (Hex)
Millivolts DC (mV)	+/- 15mV	0x0100
	+/- 50mV	0x0101
	+/- 100mV	0x0102
	+/- 150mV	0x0103
	+/- 500mV	0x0104
	0~150mV	0x0105
	0~500mV	0x0106
Millamps (mA)	4~20mA	0x0180
	+/-20mA	0x0181
	0~20mA	0x0182
Counter settings	Pulse/DIR	0x01C0
	Up/Down	0x01C1
	Up	0x01C2
	Frequency	0x01C3
	AB 1X	0x01C4
	AB 2X	0x01C5
	AB 4X	0x01C6
Pt-100 (3851)	Pt-100 (3851) -200~850 °C	0x0200
	Pt-100 (3851) -120~130 °C	0x0201
	Pt-100 (3851) -200~200 °C	0x0202
	Pt-100 (3851) -100~100 °C	0x0203
	Pt-100 (3851) -50~150 °C	0x0204
	Pt-100 (3851) 0~100 °C	0x0205
	Pt-100 (3851) 0~200 °C	0x0206
	Pt-100 (3851) 0~400 °C	0x0207
Pt-200 (3851)	Pt-200 (3851) -200~850 °C	0x0220
	Pt-200 (3851) -120~130 °C	0x0221
Pt-500 (3851)	Pt-500 (3851) -200~850 °C	0x0240
	Pt-500 (3851) -120~130 °C	0x0241
Pt-1000 (3851)	Pt-1000 (3851) -200~850 °C	0x0260
	Pt-1000 (3851) -120~130 °C	0x0261
	Pt-1000 (3851) -40~160 °C	0x0262

Appendix A APAX 5000 I/O module range code settings

Pt-100 (3916)	Pt-100 (3916) -200~850 °C	0x0280
	Pt-100 (3916) -120~130 °C	0x0281
	Pt-100 (3916) -200~200 °C	0x0282
	Pt-100 (3916) -100~100 °C	0x0283
	Pt-100 (3916) -50~150 °C	0x0284
	Pt-100 (3916) 0~100 °C	0x0285
	Pt-100 (3916) 0~200 °C	0x0286
	Pt-100 (3916) 0~400 °C	0x0287
	Pt-100 (3916) 0~600 °C	0x0288
Pt-200 (3916)	Pt-200 (3916) -200~850 °C	0x02A0
	Pt-200 (3916) -120~130 °C	0x02A1
Pt-500 (3916)	Pt-500 (3916) -200~850 °C	0x02C0
	Pt-500 (3916) -120~130 °C	0x02C1
Pt-1000 (3916)	Pt-1000 (3916) -200~850 °C	0x02E0
	Pt-1000 (3916) -120~130 °C	0x02E1
	Pt-1000 (3916) -40~160 °C	0x02E2
Balco 500	Balcon(500) -30~120	0x0300
Ni 518	Ni(518) -80~100 °C	0x0320
	Ni(518) 0~100 °C	0x0321
Ni 508	Ni(508) 0~100 °C	0x0340
	Ni(508) -50~200 °C	0x0341
Thermistor 3K	Thermistor 3K 0~100 °C	0x0360
Thermistor 10K	Thermistor 10K 0~100 °C	0x0380
	Thermistor 10K -50~100 °C	0x0381
T/C TypeJ	T/C TypeJ 0~760 °C	0x0400
	T/C TypeJ -200~1200 °C	0x0401
T/C TypeK	T/C TypeK 0~1370 °C	0x0420
	T/C TypeK -270~1372 °C	0x0421
T/C TypeT	T/C TypeT -100~400 °C	0x0440
	T/C TypeT -270~400 °C	0x0441
T/C TypeE	T/C TypeE 0~1000 °C	0x0460
	T/C TypeE -270~1000 °C	0x0461
T/C TypeR	T/C TypeR 500~1750 °C	0x0480
	T/C TypeR 0~1768 °C	0x0481
T/C TypeS	T/C TypeS 500~1750 °C	0x04A0
	T/C TypeS 0~1768 °C	0x04A1
T/C TypeB	T/C TypeB 500~1800 °C	0x04C0
	T/C TypeB 300~1820 °C	0x04C1



Enabling an Intelligent Planet

www.advantech.com

Please verify specifications before quoting. This guide is intended for reference purposes only.

All product specifications are subject to change without notice.

No part of this publication may be reproduced in any form or by any means, electronic, photocopying, recording or otherwise, without prior written permission of the publisher.

All brand and product names are trademarks or registered trademarks of their respective companies.

© Advantech Co., Ltd. 2012